Safe extensions to the stochastic analysis of real-time systems

Jóse Luis Díaz, Jóse María López
Universidad de Oviedo

Technical report SESART04
January 2004

Abstract

Classical deterministic analysis of real-time systems focuses on obtaining the worst possible response time of each task, among all possible execution scenarios. Comparing this worst-case response time with the deadline of the task, the feasibility of the system can be assessed. If the deadline does not exceed this worst-case execution time, it is guaranteed that the response time of the deadline will not exceed its deadline under any circunstance.

The stochastic analysis of real-time systems, instead, focuses in obtaining the probability function of the response time of the tasks, that is, the profile of all possible response times and the probability of occurrence of each one. Using this probability function, the probability of the response time exceeding any given value can be computed. In particular, the probability of deadline misses can be obtained, since it is equal to the probability of the response time exceeding that deadline. However, this kind of stochastic analysis has a high computational complexity, and becomes unffordable in current practice. On one side, the exact calculation of the response time distribution of the tasks is not possible except for simple periodic and independent task sets. On the other side, in practice, tasks introduce complexities like release jitter, blocking in shared resources, stochastic dependencies, etc, which can not be handled by the periodic and independent task set model.

In order to overcome the problems of the stochastic analysis, some kind of simplification has to be incorporated into it. However, it is important to guarantee that the simplifications incorporated are *safe*, or *pessimistic* in an stochastic sense. We will consider a simplification to be safe if the probabilities of deadline misses obtained using that simplification are greater than the exact ones. This way, we can guarantee that the real ratio of deadline misses will never exceed the probabilities obtained from the analysis.

This technical report introduces the idea of pessimism in the stochastic analysis, by formally defining a comparison operator among random variables, which allow us to state whether a random variable is “worse than” other. Some important properties of this operator are shown. These properties are used to prove some theorems about the stochastic analysis, which can be summarized as “if pessimistic data is introduced in the model, pessimistic response times will be obtained”. This result looks trivial, but it has to be carefully proven when the model deals with random variables instead of deterministic constants.

Allowing some pessimism, the stochastic analysis of real-time can be extended in several directions. In this technical report we present one of them, namely, the extension for dealing with non-independent tasks, which can be blocked in the access to shared resources. We will show that the classical resource sharing protocols can be also stochastically analyzed, if pessimism is allowed.

1 Introduction

Traditional techniques of real-time analysis assume single-valued execution times of the tasks. For example, the processor utilization analysis [11, 10] and response time analysis [16].

However, execution times are variable in practice, so analysis techniques based on single-valued execution times usually consider only the worst-case execution time. This greatly introduces pessimism in the analysis, giving rise to over dimensioned real-time systems.

A multiframe model was proposed by Mok and Chen [13], in which the execution time of a task may vary greatly from one instance to another, assuming that this variation follows a known pattern. The pattern is given as a finite list of numbers, and the execution times of successive instances are generated from the list. From this model, new utilization bounds which improve those of Liu and Layland [11] are derived for fixed-priority preemptive scheduling. However, since this model is aimed at providing a deterministic timing guarantee, it is still pessimistic.

Progress has recently been made in the analysis of real-time systems under the assumption that tasks require stochastic execution times. Research in this area can be categorized into two groups depending on the approach it takes to facilitate the analysis.

The methods in the first group assume a special scheduling model that provides isolation between tasks, so that each task can be analyzed independently of other tasks in the system (e.g., the reservation-based system addressed in [1] and Statistical Rate Monotonic Scheduling [2]). Those in the second group consider common scheduling algorithms and model the execution time of the tasks as random variables. They introduce worst-case assumptions to simplify the analysis (e.g., the critical instant assumption in Probabilistic Time Demand Analysis [15] and Stochastic Time Demand Analysis [6, 7]), restrictive load conditions (e.g., the heavy traffic condition in the Real-Time Queuing Theory [8, 9]), or restrictions about maximum system utilization and preemption [12].

Diaz et al. [5] modeled the execution time of the tasks as random variables and performed an exhaustive analysis of periodic and independent tasks sets without worst-case assumptions under both fixed-priority and EDF scheduling. The authors proved that the system becomes stable whenever the average system utilization is less than one. For this case, they present algorithms to calculate the stochastic distribution of the response time of the tasks, which allow us to calculate the probability of missing deadlines. An interesting property of the analysis is the capacity to deal with systems with maximum system utilization higher than one (whenever the average system utilization remains lower than one). Thus, systems which would be deemed unfeasible under the deterministic analysis may be feasible under the stochastic analysis. The analysis is useful not only for soft real-time systems, but also for the so-called *probabilistic hard real-time systems* [3], where a probabilistic guarantee close to 100% suffices.

However, the exact stochastic analysis proposed in [5] has a high computational complexity, and becomes unffordable in current practice. On one side, the exact calculation of the response time distribution of the tasks is not possible except for simple periodic and independent task sets. On the other side, in practice, tasks introduce complexities like release jitter, blocking in shared resources, stochastic dependencies, etc, which can not be handled by the periodic and independent task set model. Extending the model to cover these cases, altough possible in theory, would increase even more his computational complexity.

There is a clear need of some way to simplify the analysis and reduce the amount of data used by the algorithms. However, since the analysis is aimed to probabilistic hard real-time systems, these simplifications should be guaranteed *pessimistic*, i.e. the real system should never behave *worse* than predicted by the analysis. This technical report formalices the idea of *worse* in the stochastic sense, and proves that this idea leads to safe simplifications and extensions. It also presents one of its multiple applications, the extension of the model for dealing with blocking in the access to shared resources.

The rest of the technical report is organized as follows. In Section 2 the model of the system is presented. In Section 3 the concept of *stochastic pessimism* and the relationship "*worse than*" applied to random variables are presented, and some of its more important propierties are proved. In Section 5 the extension of the model to deal with blocking is presented. This extension uses the *stochastic pessimism* idea defined before. Finally, in Section 6 the conclusions and future work are presented.

2 System model

The system is modeled as a set of N independent periodic tasks $S = \{\tau_1, \dots, \tau_i, \dots, \tau_N\}$, each task τ_i being defined by the tuple $(T_i, \Phi_i, \mathcal{C}_i, D_i, M_i)$, where T_i is the period of the task, Φ_i its initial phase, $\mathcal{C}_i$ its execution time and the pair (D_i, M_i) define the real-time constraint of the task.

The execution time is a discrete random variable[1] with a known probability function (PF), denoted by $f_{\mathcal{C}_i}(\cdot)$, where $f_{\mathcal{C}_i}(c) = \mathbb{P}\{\mathcal{C}_i{=}c\}$. Alternatively, the execution time distribution can also be specified using its cumulative distribution function (CDF), denoted by $F_{\mathcal{C}_i}(\cdot)$, where $F_{\mathcal{C}_i}(x) = \sum_{c=0}^{x} f_{\mathcal{C}_i}(c)$. The value of $\mathcal{C}_i$ is bounded by a positive minimum value $C_i^{\min}$ and a finite maximum value $C_i^{\max}$, so its probability function can be stored as a finite vector $[f_{\mathcal{C}_i}(C_i^{\min}), \dots, f_{\mathcal{C}_i}(C_i^{\max})]$.

Three system utilizations can be derived from the execution time of the tasks, namely:

- Minimum system utilization: $U^{\min} = \sum\limits_{i=1}^{N} C_i^{\min}/T_i$
- Average system utilization: $\overline{U} = \sum\limits_{i=1}^{N} \overline{C_i}/T_i$
- Maximum system utilization: $U^{\max} = \sum\limits_{i=1}^{N} C_i^{\max}/T_i$

Each periodic task gives rise to an infinite sequence of jobs, Γ_j, with deterministic release times λ_j, which depends only on the phase and period of the task it comes from. Each job requires an execution time which is a random variable whose distribution is given by the probability function of the task it comes from, $f_{\mathcal{C}_i}(\cdot)$, and it is assumed to be independent of other jobs of the same task and those of other tasks. The response time of a job Γ_j is a random variable, denoted by $\mathcal{R}_j$, whose probability function has to be obtained by the analysis.

D_i is the task relative deadline and M_i the maximum allowable probability of missing deadline D_i. Task τ_i is said to be schedulable if $\mathbb{P}\{\mathcal{R}_i{>}D_i\} \leq M_i$, being $\mathcal{R}_i$ the response time of τ_i.

[1]Throughout this report we use a calligraphic typeface to denote random variables, e.g. $\mathcal{C}$, $\mathcal{W}$, $\mathcal{R}$, etc.

The scheduling policy we assume is a general, preemptive, priority-driven policy that assigns a static priority to each job and schedules jobs according to this priority. The scheduler guarantees that the running job is the one with the highest priority among the ready jobs. We are not concerned with the policy used to assign priorities to jobs, as long as they are assigned in a deterministic way. This model includes well-known fixed priority policies such as *Deadline Monotonic* (DM), and non-fixed priority policies such as *Earliest Deadline First* (EDF).

The probability function of the response time of a task can be theoretically obtained by averaging the response time probability functions of the jobs that the task generates. The problem is that this sequence of jobs is infinite. However, due to the periodic nature of the release pattern of the jobs it is to be expected that the probabilities of deadline misses of these jobs also exhibit a periodic pattern.

Diaz et al. [5] proved that the statistical distribution of the response time of the job released at instant λ_j is the same than that of the job released at instant $\lambda_j + T$, i.e, one hyperperiod[2] later, whenever $U^{\max} \leq 1$. Moreover, if $\overline{U} < 1$, even if $U^{\max} > 1$, the statistical distribution of the response time of the jobs released at $\lambda_j, \lambda_j + T, \lambda_j + 2T, \ldots, \lambda_j + kT$ converges towards a steady-state distribution.

This means that, in order to compute the response time probability function of a task, it suffices to obtain the response time probability functions of the jobs generated by this task within an hyperperiod. Diaz et al. [5] provided algorithms for obtaining the probability function of the initial backlog in the stady-state hyperperiod, and from this backlog the response time for all jobs in the steady-state hyperperiod can be derived.

These algorithms operate on probability functions which are assumed to be *exact*. The computational cost of these algorithms is high. If the model has to be extended for overcoming some of its limitations, the complexity will grow exponentially. There is a need for reducing this complexity, at the cost of lossing some precision in the results. However, in order to these results be useful for design decissions in the field of real-time systems, the loss of precision must be on the safe side. That is, the probabilities of deadline misses provided by the approximated solution must be greater than the exacto probabilities. In the next section we formalize the concept “worse than” which will allow us to design this kind of safe simplifications and extensions.

3 The concept of pessimism and its basic properties

It is possible to define an ordering among the random variables, such that we can say that one random variable is “worse than” other, in the context of real-time systems. This allows us, to compare different analysis techniques and approximations. We will state that a analysis technique or approximation is *pessimistic*, if the resultant response times of the tasks are stochastically worse than the real ones. The concept of “worse than” in the real-time stochastic analysis coincides with the concept of “first order stochastic dominance” introduced in statistics and further used in economics. Its formal definition is as follows:

Definition 1. *Given two random variables $\mathcal{X}$ and $\mathcal{Y}$, we will say that “$\mathcal{X}$ is worse than $\mathcal{Y}$”, and denote it by $\mathcal{X} \succcurlyeq \mathcal{Y}$ if $F_{\mathcal{X}}(x) \leq F_{\mathcal{Y}}(x)$.*

Graphically, this means that the curve $F_{\mathcal{X}}(\cdot)$ never goes above the curve $F_{\mathcal{Y}}(\cdot)$ (see fig 1). Note that if the curves $F_{\mathcal{X}}(\cdot)$ and $F_{\mathcal{Y}}(\cdot)$ cross, the variables $\mathcal{X}$ and $\mathcal{Y}$ are not comparable, and it is not true that $\mathcal{X} \succcurlyeq \mathcal{Y}$ nor $\mathcal{Y} \succcurlyeq \mathcal{X}$. In this case, all what can be stated is $\mathcal{X} \not\succcurlyeq \mathcal{Y}$

[2]The hyperperiod, T, is defined as the least common multiple of the periods of all the tasks.

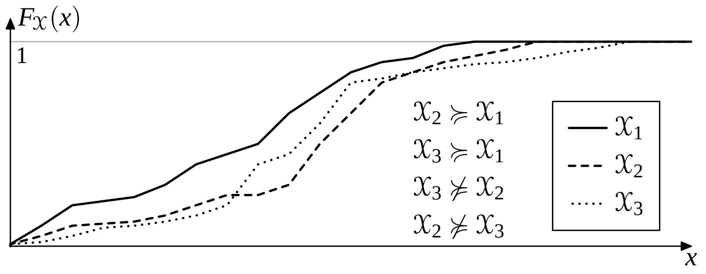

Figure 1: Graphical meaning of the relationship "worse than"

and $\mathcal{Y} \not\succcurlyeq \mathcal{X}$ (see fig. 1). The relationship "worst than" provides a partial ordering among the random variables.

This kind of ordering is useful for the stochastic analysis of real-time systems. Suppose that $\mathcal{R}'$ is the approximated response time of a task provided by an approximated stochastic analisis, while $\mathcal{R}$ is the exact response time. If we could guarantee (i.e, mathematically prove) that $\mathcal{R}' \succcurlyeq \mathcal{R}$, then the analysis would be safe, because this would imply $\mathbb{P}\{\mathcal{R}{>}D\} \leq \mathbb{P}\{\mathcal{R}'{>}D\}$, that is, for any deadline, the "true" probability of deadline misses is inferior to the probability provided by the analysis.

We will show then some general properties of the relation "worse than". Most of these are very intuitive, and their proof is very simple. However, once proved, these properties constitute a kind of "algebra" which can be used to prove important properties of the stochastic analysis method. The motivation is to provide a theoretical ground for mathematically prove wether a given approximated analysis technique is safe or not.

As it is known, the probability function of a sum of random variables can be obtained by convolution of the probability functions of these variables. However, the notion of "worse than" relies on the cumulative distribution function (CDF) and not on the probability function. Next lemma provides a way to directly obtain the CDF of a sum of random variables. This will be used in the proof of several properties.

Lemma 1. *Being $\mathcal{X}$ and $\mathcal{Y}$ two random variables whose PF are $f_{\mathcal{X}}(\cdot)$ and $f_{\mathcal{Y}}(\cdot)$ respectively, and whose CDF are $F_{\mathcal{X}}(\cdot)$ and $F_{\mathcal{Y}}(\cdot)$, respectively, the CDF of their sum can be calculated by:*

$$F_{\mathcal{X}+\mathcal{Y}}(x) = \big(F_{\mathcal{X}} \otimes f_{\mathcal{Y}}\big)(x) = \big(f_{\mathcal{X}} \otimes F_{\mathcal{Y}}\big)(x) \tag{1}$$

Proof. By definition,

$$F_{\mathcal{X}+\mathcal{Y}}(x) \triangleq \sum_{j=-\infty}^{x} f_{\mathcal{X}+\mathcal{Y}}(j) = \sum_{j=-\infty}^{x} \big(f_{\mathcal{X}} \otimes f_{\mathcal{Y}}\big)(j) \triangleq \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{x} f_{\mathcal{X}}(i) \cdot f_{\mathcal{Y}}(j-i)$$

The coefficient $f_{\mathcal{X}}(i)$ does not depend on j, so it can be taken out of the inner sum:

$$\begin{aligned}
F_{\mathcal{X}+\mathcal{Y}}(x) &= \sum_{i=-\infty}^{\infty} f_{\mathcal{X}}(i) \sum_{j=-\infty}^{x} f_{\mathcal{Y}}(j-i) && \text{taking } f_{\mathcal{X}}(i) \text{ out of the sum} \\
&= \sum_{i=-\infty}^{\infty} f_{\mathcal{X}}(i) F_{\mathcal{Y}}(x-i) && \text{by definition of } F_{\mathcal{Y}}(\cdot) \\
&= \big(f_{\mathcal{X}} \otimes F_{\mathcal{Y}}\big)(x) && \text{by definition of } \otimes
\end{aligned}$$

Moreover, since the convolution operator is conmutative, this also proves $F_{\mathcal{X}+\mathcal{Y}}(x) = \big(F_{\mathcal{X}} \otimes f_{\mathcal{Y}}\big)(x)$ □

Computing the CDF using the formulae given in this lemma is not practical. An implementation of the analysis should use the convolution of the PFs, because this convolution involves a finite number of operations (only the non-zero elements of the PF need to be convoluted). The convolution of a CDF with a PF, instead, would require an infinite number of operations, because the CDF has an infinite number of non-zero elements. However, the formulae given in lemma 1 will be useful for the proofs related with the "worse than" relation. Also note that the lemma still holds if any of the functions $f_{\mathcal{A}}(\cdot)$ or $f_{\mathcal{B}}(\cdot)$ has integral less than 1. In this case, there is a "probability deficit" which can be pictured as a pulse at the infinity, with a probability equal to the deficit. The corresponding $F_{\mathcal{A}}(\cdot)$ simply never reaches the value 1.

In order to simplify the notation, we will say that a random variable $\mathcal{X}$ is *positive*, if $\mathcal{X} \succcurlyeq \mathbb{O}$, being $\mathbb{O}$ the random variable whose probability function is defined as:

$$f_{\mathbb{O}}(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases} \tag{2}$$

Now, we can state and prove the following properties.

Property 1. *Reflexivity:* $\mathcal{A} \succcurlyeq \mathcal{A}$.

Proof. The proof is direct from definition 1. □

Property 2. *Transitivity: if* $\mathcal{A} \succcurlyeq \mathcal{B}$ *and* $\mathcal{B} \succcurlyeq \mathcal{C}$*, then* $\mathcal{A} \succcurlyeq \mathcal{C}$.

Proof. The proof is direct from definition 1. □

Property 3. *If* $\mathcal{A} \succcurlyeq \mathcal{B}$*, then for all* $\mathcal{C}$ *positive* $\mathcal{A} + \mathcal{C} \succcurlyeq \mathcal{B} + \mathcal{C}$*. That is, we can add any positive variable to both members of an inequality, without altering it.*

Proof. By lemma 1, $F_{\mathcal{A}+\mathcal{C}}(\cdot) = \big(F_{\mathcal{A}} \otimes f_{\mathcal{C}}\big)(\cdot)$, and analogously $F_{\mathcal{B}+\mathcal{C}}(\cdot) = \big(F_{\mathcal{B}} \otimes f_{\mathcal{C}}\big)(\cdot)$. Developing these convolutions into summatories, and since $F_{\mathcal{A}}(\cdot) \leq F_{\mathcal{B}}(\cdot)$ by hypothesis, we conclude that $F_{\mathcal{A}+\mathcal{C}}(\cdot) \leq F_{\mathcal{A}+\mathcal{B}}(\cdot)$, which, by definition, implies $\mathcal{A} + \mathcal{C} \succcurlyeq \mathcal{B} + \mathcal{C}$. □

Property 4. *For all positive* $\mathcal{A}, \mathcal{B}$*, it is true that* $\mathcal{A} + \mathcal{B} \succcurlyeq \mathcal{A}$ *and* $\mathcal{A} + \mathcal{B} \succcurlyeq \mathcal{B}$*. That is, the result of adding two positive random variables is always worse than any of them.*

Proof. By hipothesis $\mathcal{B} \succcurlyeq \mathbb{O}$. According with property 3 we can add a positive random variable to both members, so $\mathcal{A} + \mathcal{B} \succcurlyeq \mathcal{A} + \mathbb{O}$. It is trivial to prove that $\big(f_{\mathcal{A}} \otimes f_{\mathbb{O}}\big)(\cdot) = f_{\mathcal{A}}(\cdot)$, so $\mathcal{A} + \mathbb{O} = \mathcal{A}$, and the proof of the first inequality is complete. The second inequality is proved the same way. □

Property 5. *For any positive* $\mathcal{A}, \mathcal{B}, \mathcal{C}$ *and* $\mathcal{D}$ *such that* $\mathcal{A} \succcurlyeq \mathcal{B}$ *and* $\mathcal{C} \succcurlyeq \mathcal{D}$*, it holds that* $\mathcal{A} + \mathcal{C} \succcurlyeq \mathcal{B} + \mathcal{D}$*. That is, we can add two inequalities member by member.*

Proof. By hipothesys, $\mathcal{A} \succcurlyeq \mathcal{B}$, and since $\mathcal{C}$ is positive, $\mathcal{A} + \mathcal{C} \succcurlyeq \mathcal{B} + \mathcal{C}$ by property 3. Analogously, since $\mathcal{C} \succcurlyeq \mathcal{D}$ and $\mathcal{B}$ is positive, we have $\mathcal{B} + \mathcal{C} \succcurlyeq \mathcal{B} + \mathcal{D}$. By the transitive property, we conclude that $\mathcal{A} + \mathcal{C} \succcurlyeq \mathcal{B} + \mathcal{D}$. □

4 Properties of the stochastic analysis

Using the above general properties, we can prove now some properties of the stochastic analysis in [5]. In order to simplify the notation and the proofs, we will reformulate as mathematical functions some of the algorithms presented in [5].

The algorithm "*convolve and shrink*" can be expressed as an iterative formula in the following way:

$$\begin{aligned} &\mathcal{W}(\lambda_1) = \mathbb{O} \\ &\mathcal{W}(\lambda_j) = \text{SHRINK}(\mathcal{W}(\lambda_{j-1}) + \mathcal{C}_{j-1}, \lambda_j - \lambda_{j-1}) \quad \text{for } j > 1 \end{aligned} \tag{3}$$

where $\text{SHRINK}(\mathcal{W}, \Delta)$ is a function, which takes the random variable $\mathcal{W}$ and the integer Δ and produces a new random-variable whose probability function is equal to the probability function of $\mathcal{W}$, left-shifted the amount Δ and with all values for negative abscissae accumulated at zero. That is:

$$f_{\text{SHRINK}(\mathcal{W},\Delta)}(x) = \begin{cases} 0 & \text{if } x < 0 \\ \sum_{i=-\infty}^{0} f_{\mathcal{W}}(x+\Delta) & \text{if } x = 0 \\ f_{\mathcal{W}}(x+\Delta) & \text{if } x > 0 \end{cases} \tag{4}$$

The algorithm "*split, convolve and merge*" can be expressed as an iterative formula in the following way:

$$\begin{aligned} &\mathcal{R}_j^{\langle 0 \rangle} = \mathcal{W}(\lambda_j) + \mathcal{C}_j \\ &\mathcal{R}_j^{\langle k+1 \rangle} = \text{CF}(\mathcal{R}_j^{\langle k \rangle}, \lambda_k - \lambda_j, \mathcal{C}_k) \quad \text{for } k > 0 \end{aligned} \tag{5}$$

where $\text{CF}(\mathcal{R}, \Delta, \mathcal{C})$ is the function "convolve from", which takes two random variables $\mathcal{R}$ and $\mathcal{C}$, and the integer Δ, and produces a new random variable whose probability function is equal to

$$f_{\text{CF}(\mathcal{R},\Delta,\mathcal{C})}(x) = \begin{cases} f_{\mathcal{R}}(x) & \text{for } x \leq \Delta \\ \sum_{i=\Delta+1}^{\infty} f_{\mathcal{R}}(i) \cdot f_{\mathcal{C}}(x-i) & \text{for } x > \Delta \end{cases} \tag{6}$$

The following lemma provides a way for obtaining the cummulative distribution function (CDF) of the random variable $\text{CF}(\mathcal{R}, \Delta, \mathcal{C})$. This will be useful to prove some properties of the CF function.

Lemma 2. *The cummulative probability function $F_{\text{CF}(\mathcal{R},\Delta,\mathcal{C})}(x)$ can be calculated with the following expression:*

$$F_{\text{CF}(\mathcal{R},\Delta,\mathcal{C})}(x) = \begin{cases} F_{\mathcal{R}}(x) & \textit{for } x \leq \Delta \\ F_{\mathcal{R}}(\Delta) + F_{\widehat{\mathcal{R}}+\mathcal{C}}(x) & \textit{for } x > \Delta \end{cases} \tag{7}$$

Being $\widehat{\mathcal{R}}$ the random variable whose probability function is:

$$f_{\widehat{\mathcal{R}}}(x) = \begin{cases} 0 & \textit{if } x \leq \Delta \\ f_{\mathcal{R}}(x) & \textit{if } x > \Delta \end{cases} \tag{8}$$

Proof. Looking at the definition of $f_{\mathrm{CF}(\mathcal{R},\Delta,\mathcal{C})}(\cdot)$ in eq.(6), we can see that the second case is like a convolution, but the sum index starts from $i = \Delta + 1$ instead of $i = -\infty$. However, if we change $f_{\mathcal{R}}(x)$ by $f_{\widehat{\mathcal{R}}}(x)$ in the sum, the result will be the same, and the sum index can now be extended to $i = -\infty$ without altering the result. This second case can thus be rewritten as $f_{\mathrm{CF}(\mathcal{R},\Delta,\mathcal{C})}(x) = \big(f_{\widehat{\mathcal{R}}} \otimes f_{\mathcal{C}}\big)(x)$, for $x > \Delta$.

By definition, $F_{\mathrm{CF}(\mathcal{R},\Delta,\mathcal{C})}(x) = \sum\limits_{i=-\infty}^{x} f_{\mathrm{CF}(\mathcal{R},\Delta,\mathcal{C})}(x)$. It is not difficult to see, using the same techniques than in the proof of lemma 1, that this CDF can be computed by

$$F_{\mathrm{CF}(\mathcal{R},\Delta,\mathcal{C})}(x) = \begin{cases} F_{\mathcal{R}}(x) & \text{for } x \leq \Delta \\ F_{\mathcal{R}}(\Delta) + \big(F_{\widehat{\mathcal{R}}} \otimes f_{\mathcal{C}}\big)(x) & \text{for } x > \Delta \end{cases} \tag{9}$$

And, since $\big(F_{\widehat{\mathcal{R}}} \otimes f_{\mathcal{C}}\big)(x) = F_{\widehat{\mathcal{R}}+\mathcal{C}}(x)$, the proof of the lemma is complete. □

Next, we prove some properties of the functions SHRINK() and CF() defined above, related with the pessimism concept.

Property 6. *For any* $\mathcal{A} \succcurlyeq \mathcal{B}$, $\Delta \geq 0$, *it holds that* $\mathrm{SHRINK}(\mathcal{A},\Delta) \succcurlyeq \mathrm{SHRINK}(\mathcal{B},\Delta)$.

Proof. Let us call $\mathcal{A}'$ and $\mathcal{B}'$ to the "shrinked" versions of $\mathcal{A}$ and $\mathcal{B}$, respectively. By definition, $F_{\mathcal{A}'}(x)$ is equal to zero for $x < 0$, and equal to $F_{\mathcal{A}}(x+\Delta)$, for $x \geq 0$. In addition, by hypothesis, $F_{\mathcal{A}}(x) \leq F_{\mathcal{B}}(x)$ for all x, so, trivially, $F_{\mathcal{A}'}(x) \leq F_{\mathcal{B}'}(x)$ for all x. □

Property 7. *The operator "convolve from" defined in eq. (6), has the following properties:*

a) If $\mathcal{C}_1 \succcurlyeq \mathcal{C}_2$, *then* $\mathrm{CF}(\mathcal{R},\Delta,\mathcal{C}_1) \succcurlyeq \mathrm{CF}(\mathcal{R},\Delta,\mathcal{C}_2)$

b) If $\mathcal{R}_1 \succcurlyeq \mathcal{R}_2$, *then* $\mathrm{CF}(\mathcal{R}_1,\Delta,\mathcal{C}) \succcurlyeq \mathrm{CF}(\mathcal{R}_2,\Delta,\mathcal{C})$

c) If $\Delta \geq 0$ *and* $\mathcal{C} \succcurlyeq \mathbb{O}$, *then* $\mathrm{CF}(\mathcal{R},\Delta,\mathcal{C}) \succcurlyeq \mathcal{R}$

d) If $\Delta_1 \leq \Delta_2$, *then* $\mathrm{CF}(\mathcal{R},\Delta_1,\mathcal{C}) \succcurlyeq \mathrm{CF}(\mathcal{R},\Delta_2,\mathcal{C})$

Proof.

(a) For $x \leq \Delta$, looking at the first case in eq. (7), since it does not depend on $\mathcal{C}_1$ nor $\mathcal{C}_2$, it is clear that $F_{\mathrm{CF}(\mathcal{R},\Delta,\mathcal{C}_1)}(x) = F_{\mathrm{CF}(\mathcal{R},\Delta,\mathcal{C}_2)}(x)$.

For $x > \Delta$, by hyphotesis $\mathcal{C}_1 \succcurlyeq \mathcal{C}_2$, so, by property 3, $\widehat{\mathcal{R}} + \mathcal{C}_1 \succcurlyeq \widehat{\mathcal{R}} + \mathcal{C}_2$, which implies by definition $F_{\widehat{\mathcal{R}}+\mathcal{C}_1}(x) \leq F_{\widehat{\mathcal{R}}+\mathcal{C}_2}(x)$, and thus $F_{\mathrm{CF}(\mathcal{R},\Delta,\mathcal{C}_1)}(x) \leq F_{\mathrm{CF}(\mathcal{R},\Delta,\mathcal{C}_2)}(x)$, by lemma 2.

(b) For $x \leq \Delta$, since by hyphotesis $F_{\mathcal{R}_1}(\cdot) \leq F_{\mathcal{R}_2}(\cdot)$, from the first case in eq. (7) it follows that $F_{\mathrm{CF}(\mathcal{R}_1,\Delta,\mathcal{C})}(x) \leq F_{\mathrm{CF}(\mathcal{R}_2,\Delta,\mathcal{C})}(x)$.

For $x > \Delta$, by hypothesis $\mathcal{R}_1 \succcurlyeq \mathcal{R}_2$, so also $\widehat{\mathcal{R}}_1 \succcurlyeq \widehat{\mathcal{R}}_2$, and then $\widehat{\mathcal{R}}_1 + \mathcal{C} \succcurlyeq \widehat{\mathcal{R}}_2 + \mathcal{C}$, which implies $F_{\widehat{\mathcal{R}}_1+\mathcal{C}}(x) \leq F_{\widehat{\mathcal{R}}_2+\mathcal{C}}(x)$. Thus, $F_{\mathrm{CF}(\mathcal{R}_1,\Delta,\mathcal{C})}(x) \leq F_{\mathrm{CF}(\mathcal{R}_2,\Delta,\mathcal{C})}(x)$, by lemma 2.

(c) It is trivial to prove that $\mathcal{R} = \mathrm{CF}(\mathcal{R},\Delta,\mathbb{O})$, and then, by property (a), the property (c) follows.

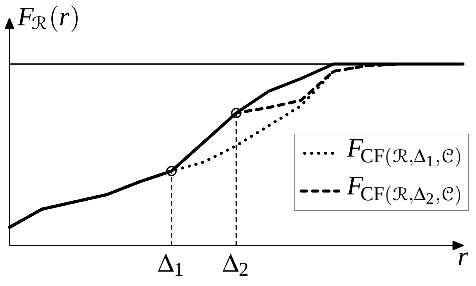

Figure 2: Effect of different Δ on the CF operator (property 7d)

(d) Consider first the case $x \le \Delta_2$. On one hand, according to eq. (6), $F_{\mathrm{CF}(\mathcal{R},\Delta_2,\mathcal{C})}(x) = F_{\mathcal{R}}(x)$. on the other hand, for any x, $F_{\mathrm{CF}(\mathcal{R},\Delta_1,\mathcal{C})}(x) \le F_{\mathcal{R}}(x)$, because $\mathrm{CF}(\mathcal{R},\Delta_1,\mathcal{C}) \succcurlyeq \mathcal{R}$, as demonstrated in property (c) (cf. figure 2). So we have $F_{\mathrm{CF}(\mathcal{R},\Delta_1,\mathcal{C})}(x) \le F_{\mathrm{CF}(\mathcal{R},\Delta_2,\mathcal{C})}(x)$ for $x \le \Delta_2$.

The case $x > \Delta_2$ is slightly more complex. Starting from eq.(7)), by lemma 1 we can compute $F_{\widehat{\mathcal{R}}+\mathcal{C}}(\cdot)$ as $\left(f_{\widehat{\mathcal{R}}} \otimes F_{\mathcal{C}}\right)(\cdot) = \sum_{i=-\infty}^{\infty} \widehat{f_{\mathcal{R}}}(i) \cdot F_{\mathcal{C}}(x-i)$. Noting that $\widehat{f_{\mathcal{R}}}(i)$ is zero for $i \le \Delta$, and equal to $f_{\mathcal{R}}(i)$ for $i > \Delta$, we can finally write:

$$F_{\mathrm{CF}(\mathcal{R},\Delta_1,\mathcal{C})}(x) = F_{\mathcal{R}}(\Delta_1) + \sum_{i=\Delta_1+1}^{\infty} f_{\mathcal{R}}(i) \cdot F_{\mathcal{C}}(x-i) \tag{10}$$

$$F_{\mathrm{CF}(\mathcal{R},\Delta_2,\mathcal{C})}(x) = F_{\mathcal{R}}(\Delta_2) + \sum_{i=\Delta_2+1}^{\infty} f_{\mathcal{R}}(i) \cdot F_{\mathcal{C}}(x-i) \tag{11}$$

The sum in eq. (10) can be split in two sums, one from $i = \Delta_1 + 1$ to Δ_2, and other from $i = \Delta_2 + 1$ to ∞. Then, substracting eq. (10) from eq. (11) we obtain

$$F_{\mathrm{CF}(\mathcal{R},\Delta_2,\mathcal{C})}(x) - F_{\mathrm{CF}(\mathcal{R},\Delta_1,\mathcal{C})}(x) = F_{\mathcal{R}}(\Delta_2) - F_{\mathcal{R}}(\Delta_1) - \sum_{i=\Delta_1+1}^{\Delta_2} f_{\mathcal{R}}(i) F_{\mathcal{C}}(x-i)$$

then, by definition of $F_{\mathcal{R}}(\cdot)$

$$= \sum_{i=\Delta_1+1}^{\Delta_2} f_{\mathcal{R}}(i) - \sum_{i=\Delta_1+1}^{\Delta_2} f_{\mathcal{R}}(i) F_{\mathcal{C}}(x-i)$$

and finally, taking common factor of $f_{\mathcal{R}}(i)$

$$= \sum_{i=\Delta_1+1}^{\Delta_2} f_{\mathcal{R}}(i)(1 - F_{\mathcal{C}}(x-i))$$

This sum non-negative, because $F_{\mathcal{C}}(\cdot) \le 1$, so $F_{\mathrm{CF}(\mathcal{R},\Delta_1,\mathcal{C})}(x) \le F_{\mathrm{CF}(\mathcal{R},\Delta_1,\mathcal{C})}(x)$ also for $x > \Delta_2$, q.e.d

□

The above properties have important implications in the stochastic analysis. The main idea can be informally stated as "pessimistic input data will produce pessimistic results". This idea seems obvious when the input data managed by the analysis algorithms are deterministic numbers. We prove that it is also true when the input data are random variables.

Theorem 1. *Lets S and S' be two systems with identical parameters, but with different initial backlog $\mathcal{W}(0)$ and $\mathcal{W}'(0)$ respectively. If $\mathcal{W}'(0) \succcurlyeq \mathcal{W}(0)$, then $\mathcal{W}'(t) \succcurlyeq \mathcal{W}(t)$ for all $t \geq 0$.*

Proof. Lets t_1 be equal to the arrival instant of the next job which contributes to the backlog, since S and S' have the same parameters, this instant is the same for both. For all $t < t_1$, the backlog at instant t is obtained simply as $\mathrm{SHRINK}(\mathcal{W}(0), t)$, so, by property 6, $\mathcal{W}'(t) \succcurlyeq \mathcal{W}(t)$ for $t < t_1$.

At instant $t = t_1$, the backlog is increased by the computation time of the arriving job. Lets $\mathcal{C}$ be the random variable which represents this computation time, which is the same for both systems S and S'. By property 3, $\mathcal{W}'(t_1) + \mathcal{C} \succcurlyeq \mathcal{W}(t_1) + \mathcal{C}$. Taking t_1 as the new time origin, the same reasoning can be repeated until reaching any future instant. □

This theorem implies that, when $\mathcal{W}(0) = \mathbb{O}$, the backlog "worsens" with time. It also implies that, when a new job is added to the system, the backlog worsens from the instant of relase of this new job, in relation with the case in which this job was not present. Finally, it also implies that, if the execution time of a job is replaced by one "worse" (in the stochastic sense), the backlog from the release instant of that job will be also worse.

Theorem 2. *Let S and S' be two real-time systems, with identical parameters, except for one of the jobs, say Γ_k, whose execution time is $\mathcal{C}_k$ in system S and $\mathcal{C}'_k$ in system S'. If $\mathcal{C}'_k \succcurlyeq \mathcal{C}_k$, then the response times obtained by the stochastic analysis of these systems fulfil $\mathcal{R}'_j \succcurlyeq \mathcal{R}_j$ for all Γ_j.*

Proof. All jobs with priority greater than P_k are unnafected by Γ_k, so their response time remain unnafected, and trivially $\mathcal{R}'_j \succcurlyeq \mathcal{R}_j$ for these jobs (because the "worse than" relation is reflexive). So we will focus only on jobs with priority less than P_k. Let us consider an arbitrary job Γ_j.

If $j < k$, we have that the backlog for any instant prior to λ_k is the same for both systems, because the sequence of arrivals and the execution times are the same. As a consequence, if job Γ_j cannot be preempted by Γ_k, the response time will be the same for both systems. If Γ_j can be preempted by Γ_k, then $\mathcal{R}'_j \succcurlyeq \mathcal{R}_j$, by the hypothesis $\mathcal{C}'_k \succcurlyeq \mathcal{C}_k$ and property 7(a).

If $j = k$, we are calculating the response time of job Γ_k. As in the previous case, the backlog is the same in both systems. But the response time $\mathcal{R}'_k$ will be worse than $\mathcal{R}_k$, because $\mathcal{C}'_k \succcurlyeq \mathcal{C}_k$, in virtue of properties 7(a) and 7(b) we will have $\mathcal{R}'_k \succcurlyeq \mathcal{R}_k$.

Finally, if $j > k$, we have that $\mathcal{W}'(\lambda_j) \succcurlyeq \mathcal{W}(\lambda_j)$, because at instant λ_k the backlog $\mathcal{W}$ is increased by $\mathcal{C}_k$, while the backlog $\mathcal{W}'$ is increased in $\mathcal{C}'_k$, being $\mathcal{C}'_k \succcurlyeq \mathcal{C}_k$. In virtue of Theorem 1, the backlog will be worse for any fugure instant. Then, for any job released after Γ_k, the initial backlog is worse, and thus, by properties 7(a) and 7(b) again, we conclude $\mathcal{R}'_j \succcurlyeq \mathcal{R}_j$. □

This theorem implies that, if required, the analyst can replace the computation time of any job by one more pessimistic in the stochastic sense. The results obtained after this replacement are pessimistic, but safe. This mechanism will allow for introducing several simplifications and extensions in the model.

As a corollary, introducing artificially a new job (or task) will also increase the pessimism of the results, in the sense that the response times obtained with this extra job are worse (in the stochastic sense) than without it. This can be easily proved by assuming that system S has an additional job with $\mathcal{C} = \mathbb{O}$ (which does not alter the analysis), while system S' has the same job with $\mathcal{C}' \succcurlyeq \mathbb{O}$. Now, Theorem 2 can be directly applied.

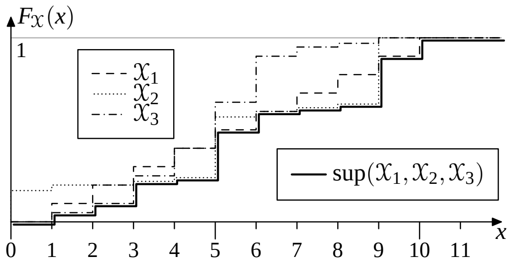

Figure 3: Construction of the supremum of a set of random variables

5 Blocking in shared resources

Shared resources, like shared memory areas, are useful to communicate tasks. Resource access protocols are used to preserve the consistency of the shared data, guaranteeing at the same time bounded blocking times. Examples of these protocols are the Priority Inheritance Protocol (PIP) and Priority Ceiling Protocol (PCP) for fixed priority scheduling, as well as the Stack Resource Protocol (SRP) for fixed and non-fixed priority scheduling [4].

Under a deterministic analysis, the response time of a task τ_i that can suffer blocking is calculated by artificially increasing its execution time by B_i units, where B_i is the blocking time of the task [4]. Since the exact blocking time can vary between different releases of the same task or be difficult to calculate, B'_i is used instead of B_i, where B'_i is a bound on the exact blocking time, B_i.

Under the stochastic analysis the situation is analogous. The execution time of a task τ_i, of execution time $\mathcal{C}_i$ should be increased by adding the blocking time $\mathcal{B}_i$, which is now a random variable. The result is a transformed task with execution time $\mathcal{C}_i + \mathcal{B}_i$ (being $f_{\mathcal{C}_i+\mathcal{B}_i} = f_{\mathcal{C}_i} \otimes f_{\mathcal{B}_i}$). Now the problem is analogous to that found in the deterministic analysis, i.e., how to calculate the exact distribution of the random variable $\mathcal{B}_i$. The solution is to find a bound valid for all scenarios. In stochastic terms, this means finding a random variable $\mathcal{B}'_i$, worse than the exact $\mathcal{B}_i$ for all possible scenarios. Let us define how to construct a random variable worse than any of a set of random variables.

Definition 2. *Given a set of random variables $\{\mathcal{X}_i\}$, we define the* supremum *of that set, and denote it as* $\sup\{\mathcal{X}_i\}$, *the random variable whose CDF is*

$$F_{\sup\{\mathcal{X}_i\}}(x) = \min_i F_{\mathcal{X}_i}(x) \tag{12}$$

Figure 3 shows how the function is constructed[3], by taking the minimum of all $F_{\mathcal{X}_i}(\cdot)$. By construction, $\sup\{\mathcal{X}_i\} \succcurlyeq \mathcal{X}_i$ for all i. Thus, the idea of the supremum of a set of random variables is analogous to the maximum of a set of real numbers. Using this idea, the classical results for resource access protocols can be easily translated to the stochastic analysis as well.

The system model has to be extended to hold information about the set of semaphores (S_k) used by the system to guard the shared resources, and the length of the critical sections in the tasks. In the classical analysis, the typical information stored in the model consists of a set of real numbers $D_{i,k}$ which represent, for each pair (τ_i, S_k), the length of the longest critical section that task τ_i contains, guarded by semaphore S_k. From this set of real numbers, the maximum blocking time B'_i of each task is obtained. The way in which

[3]Please, note that the plot of the supremum has been slightly shifted down for better legibility of the figure.

	$S_1(P_1)$	$S_2(P_1)$
τ_1	$\mathcal{D}_{1,1}$	$\mathcal{D}_{1,2}$
τ_2	$\mathcal{D}_{2,1}$	$\mathcal{D}_{2,2}$
τ_3	$\mathcal{D}_{3,1}$	$\mathcal{D}_{3,2}$

Table 1: Cumulative distributions of the critical sections for the blocking examples.

this is done depends on the resource sharing protocol. For example, PCP guarantees that each task is blocked only once by any task of less priority, while PIP only guarantees that a task cannot be blocked twice by the same semaphore or by the same task. These different properties lead to different algorithms to obtain B'_i.

These ideas can be translated to the stochastic case, using the concept of *supremum* defined above. The length of a critical section is now a random variable whose probability function is assumed known (it can be obtained by measurement, or using hybrid techniques such as the ones described in [3]). Then, a random variable $\mathcal{D}_{i,k}$ is constructed as the supremum of the length of all the critical sections of task τ_i guarded by semaphore S_k. From these $\mathcal{D}_{i,k}$ an estimation $\mathcal{B}'_i$ of the blocking time $\mathcal{B}_i$ of each task can be obtained. Once $\mathcal{B}'_i$ is obtained, the stochastic analysis can be done as described in Diaz et al. [5], but using $(\mathcal{B}'_i + \mathcal{C}_i)$ instead of $\mathcal{C}_i$. If we can guarantee that $\mathcal{B}'_i \succcurlyeq \mathcal{B}_i$, then the analysis will be pessimistic, because $\mathcal{B}'_i + \mathcal{C}_i \succcurlyeq \mathcal{B}_i + \mathcal{C}_i$, from property 3. Thus, by Theorem 2, the response times will be also pessimistic.

5.1 Priority Ceiling Protocol

It is well known that, under PCP, a task τ_i can be blocked only once by tasks of lower priority. This property ensures that the maximum blocking time B'_i that a task τ_i can suffer coincides with the length of the longest critical section among all the lower priority tasks which can cause blocking to τ_i. Translating the deterministic method presented in [14] to the stochastic case, we will compute $\mathcal{B}'_i$ as the supremum of the set $\{\mathcal{D}_{j,k} | P_j < P_i, C(S_k) \geq P_i\}$, $C(S_k)$ being the priority ceiling of the semaphore S_k, defined as the highest priority among the tasks which use that semaphore.

For example, consider a system with three tasks and two semaphores (see Table 1). The priority ceiling of each semaphore is indicated in parentheses. Each cell in the table contains the length of the critical section of task τ_i guarded by semaphore S_k. If a task τ_i does not have any critical section guarded by semaphore S_k, then $\mathcal{D}_{i,k} = \mathbb{O}$. If a task τ_i contains several sections guarded by S_k, $\mathcal{D}_{i,k}$ is computed as the supremum of the lengths of these sections, as explained before.

Task τ_1 can be blocked by any of the critical sections in lower priority tasks, because the priority ceiling of both semaphores is P_1. Then, $\mathcal{B}'_1 = \sup\{\mathcal{D}_{2,1}, \mathcal{D}_{2,2}, \mathcal{D}_{3,1}, \mathcal{D}_{3,2}\}$. The same applies to task τ_2, so $\mathcal{B}'_2 = \sup\{\mathcal{D}_{3,1}, \mathcal{D}_{3,2}\}$. Task τ_3 cannot suffer blocking because it is the lowest priority task, so $\mathcal{B}'_3 = \mathbb{O}$. Once all $\mathcal{B}'_i$ have been obtained this way, they are added to the corresponding $\mathcal{C}_i$ and the stochastic analysis is carried out as explained in [5].

5.2 Priority Inheritance Protocol

If the resource sharing protocol is PIP instead of PCP, the method for obtaining $\mathcal{B}'_i$ is different. Under PIP it has been proved [14] that any task τ_i will be blocked once at most by the same semaphore S_k, or by the same lower priority task τ_j ($P_j < P_i$). However, it is possible that the task gets blocked several times on different semaphores and by different lower priority tasks. In these cases, the blocking time will be the sum of the lengths of the critical sections which caused blocking. This implies that $\mathcal{B}'_i$ should be computed by examining all possible blocking scenarios, and taking the worst (supremum) of all these blocking times.

For instance, consider again the example in Table 1. Task τ_1 can be blocked by tasks τ_2 and τ_3, but not by the same semaphore twice. Therefore, if it gets blocked by τ_2 on semaphore S_1, task τ_3 can only cause additional blocking on semaphore S_2, and vice-versa. As a consequence, $\mathcal{B}'_1 = \sup\{(\mathcal{D}_{2,1} + \mathcal{D}_{3,2}), (\mathcal{D}_{2,2} + \mathcal{D}_{3,1})\}$. Note that, since we are dealing with random variables, each sum requires a convolution of the probability functions. Task τ_2 can be blocked only by τ_3, because τ_3 is the only lower priority task. But, since it cannot be blocked twice by the same task, we conclude $\mathcal{B}'_2 = \sup\{\mathcal{D}_{3,1}, \mathcal{D}_{3,2}\}$. Finally, task τ_3 cannot suffer blocking by lower priority tasks, so $\mathcal{B}'_3 = \mathbb{O}$.

An exhaustive analysis of all blocking scenarios is not difficult to perform, altough its computational cost can be high. The exhaustive analysis, however, can be completely avoided using a different method, at the cost of introducing even more pessimism. This method is the stochastic counterpart of the one presented in [4] for the deterministic case. In order to obtain a more pessimistic approximation, $\mathcal{B}''_i$, of the blocking time the following steps should be performed:

- For each j such that $P_j < P_i$, compute the supremum of the set $\{\mathcal{D}_{j,k} | C(S_k) \geq P_i\}$. Add all these supremi and call the result $\mathcal{B}l_i$.

- For each k such that $C(S_k) \geq P_i$, compute the supremum of the set $\{\mathcal{D}_{j,k} | P_j < P_i\}$. Add all these supremi and call the result $\mathcal{B}s_i$.

- Construct the random variable $\mathcal{B}''_i$ as one whose CDF is $F_{\mathcal{B}''_i}(x) = \max\{F_{\mathcal{B}l_i}(x), F_{\mathcal{B}s_i}(x)\}$. This concept is the inverse of the concept of supremum defined before, so we call it the *infimum*. Thus, $\mathcal{B}''_i = \inf\{\mathcal{B}l_i, \mathcal{B}s_i\}$.

Applying this approximation to the example on Table 1, $\mathcal{B}''_1 = \inf\big\{(\sup\{\mathcal{D}_{2,1}, \mathcal{D}_{2,2}\} + \sup\{\mathcal{D}_{3,1}, \mathcal{D}_{3,2}\}), (\sup\{\mathcal{D}_{2,1}, \mathcal{D}_{3,1}\} + \sup\{\mathcal{D}_{2,2}, \mathcal{D}_{3,2}\})\big\}$.

This method uses the function inf, which is the contrary than the sup. At first sight it could appear as if this will not preserve the pessimism in the results. The following theorem shows that, in fact, $\mathcal{B}''_i \succcurlyeq \mathcal{B}'_i$.

Theorem 3. *Let $\mathcal{B}'_i$ be the blocking time obtained by exhaustively analyzing all possible blocking scenarios for a task τ_i, computing the blocking time in each, and taking the supremum of all of them. Let $\mathcal{B}''_i$ be the approximation to the blocking time obtained by the method described above. Then, $\mathcal{B}''_i \succcurlyeq \mathcal{B}'_i$.*

Proof. Let us call Ω to the set of the durations of each critical sections which could cause blocking to τ_i in different scenarios. I.e, the set of durations of any critical section in tasks with priority lower than P_i, guarded by semaphores with priority ceiling greater than or equal to P_i. Formally:

$$\Omega = \{\mathcal{D}_{j,k} | P_j < P_i, C(S_k) \geq P_i\} \tag{13}$$

Let 2^Ω be the power set of Ω, i.e., the set of all its possible subsets. Each element in 2^Ω is a set which represents a blocking scenario (in which task τ_i suffers interference from all of $\mathcal{D}_{j,k}$ in that set). However, some of these sets are to be excluded, because they represent scenarios in which the properties of the PIP are violated. We will say that the set G, $G \in 2^\Omega$ represents an illegal blocking scenario if it contains at least two elements $\mathcal{D}_{j,k}, \mathcal{D}_{l,m}$ such that $j = l$ or $k = m$, because this would imply two blockings caused by the same task ($j = l$) or by the same semaphore ($k = m$).

An algorithm which computes $\mathcal{B}'_i$ by exhaustive search should do the following steps:

1. Construct the set Ω
2. For each $G \in 2^\Omega$ do:
   - If G represents a legal scenario, compute
   $$\mathcal{B}_G = \sum_{\mathcal{D}_{i,k} \in G} \mathcal{D}_{i,k} \tag{14}$$
   - If G represents an illegal scenario, $\mathcal{B}_G = \mathbb{O}$
3. Compute $\mathcal{B}'_i$ has the supremum of all $\mathcal{B}_G$ calculated before. That is:
$$\mathcal{B}'_i = \sup_{G \in 2^\Omega} \{\mathcal{B}_G\} \tag{15}$$

If we picture the random variables $\mathcal{D}_{j,k}$ arranged in an array, like the one presented in table 1, each row i will contain all the critical sections of task τ_i, while each column k will contain all the critical sections guarded by semaphore k. Any set $G \in 2^\Omega$ contains a subset of these $\mathcal{D}_{j,k}$, but if G has to represent a legal blocking scenario, then G will not contain two $\mathcal{D}_{i,j}$ in the same row, or in the same column.

On the other side, the approximated algorithm described in page 12 computes first $\mathcal{B}l_i$, as the sum of all the supremi among each row, and $\mathcal{B}s_i$ as the sum of all the supremi among each column, and finally takes the "best" (infimum) of both. We will prove that any of $\mathcal{B}l_i$ or $\mathcal{B}s_i$ is worse than $\mathcal{B}'_i$, so even if we take the "best" of them, the result will be still worse than $\mathcal{B}'_i$.

Let us consider any set G in 2^Ω, which represents a valid scenario. As said, this set will have *at maximum* one element of each row. Let us suppose the worst case in which all rows contribute to G, each with a different $\mathcal{D}_{j,k}$. By definition, $\mathcal{B}_G$ is the sum of all these $\mathcal{D}_{i,j}$. But $\mathcal{B}l_i$ is worse, because it is the sum of the supremi of each row. This supremi is, by definition, worse than any of the elements of the row, so in particular is worse than the element which contributes to $\mathcal{B}_G$. So the sum of all of them will also be worse, by property 5, i.e $\mathcal{B}l_i \succcurlyeq \mathcal{B}_G$. Since the above is true for any $G \in 2^\Omega$, it is also true for the worst (supremum) of all $\mathcal{B}_G$. This, by definition, is $\mathcal{B}'_i$ as computed by the algorithm previously described. So $\mathcal{B}l_i \succcurlyeq \mathcal{B}'_i$.

With an analogous reasoning it can be proved that $\mathcal{B}s_i \succcurlyeq \mathcal{B}'_i$. Both conclusions can be written in terms of the respective cumulative distribution functions, that is:

$$F_{\mathcal{B}l_i}(x) \leq F_{\mathcal{B}'_i}(x)$$
$$F_{\mathcal{B}s_i}(x) \leq F_{\mathcal{B}'_i}(x)$$

Since $\mathcal{B}''_i$ is the infimum of $\mathcal{B}l_i$ and $\mathcal{B}s_i$, which is built by taken the maximum, point by point, among $F_{\mathcal{B}l_i}(x)$ and $F_{\mathcal{B}l_i}(x)$, we conclude that $F_{\mathcal{B}''_i}(x) \leq F_{\mathcal{B}'_i}(x)$ which by definition implies $\mathcal{B}''_i \succcurlyeq \mathcal{B}'_i$. $\square$

6 Conclusions and future work

This report has introduced the relation *worse than* between two distributions of a random parameter of the analysis, which defines a stochastic ordering in the context of real-time systems. This relation and its properties define a theoretical framework that opens the door to safe stochastic analysis approximations. Whenever the distribution of a parameter of the stochastic analysis is substituted by a worse distribution, the resultant response time distributions coming from the analysis are worse for all the tasks, i.e, the probabilities of missing deadlines are higher and so the analysis becomes safe.

The most interesting characteristic of the relation *worse than* is that it allows us to order different distributions of the same random parameter of the analysis. For example, it allows us to state that one execution time distribution is worse than another, that a blocking time distribution is worse than another, that a response time distribution is worse than another, etc. The ordering between random variables is a valuable tool that permits a rapid translation of well-known real-time deterministic results to the stochastic scenario. Deterministic analysis becomes a particular case of the stochastic analysis. This way, any deterministic analysis is always more pessimistic than its stochastic counterpart. Using these translations we have introduced the analysis of task sets that can block on shared resources, but many others are possible.

Future work will focus on applying the pessimistic analysis to deal safely with other real-time problems, such as optimal priority assignment, release jitter or stochastic dependencies.

References

[1] L. Abeni and G. Buttazzo. Stochastic Analysis of a Reservation Based System. In *Proc. of the 9th International Workshop on Parallel and Distributed Real-Time Systems*, Apr. 2001.

[2] A. K. Atlas and A. Bestavros. Statistical Rate Monotonic Scheduling. In *Proc. of the 19th IEEE Real-Time Systems Symposium*, pages 123–132, Dec. 1998.

[3] G. Bernat, A. Colin, and S. Petters. WCET Analysis of Probabilistic Hard Real-Time Systems. In *Proc. of the 23rd IEEE Real-Time Systems Symposium*, Dec. 2002.

[4] G.C. Buttazzo. *Hard Real-Time Computing Systems. Predictable Scheduling Algorithms and Applications*, chapter 7. Kluwer Academic Publishers, Boston/Dordrecht/London, 1997.

[5] J. L. Díaz, D. F. García, Kanghee Kim, Chang-Gun Lee, Lucia Lo Bello, J. M. López, Sang Lyul Min, and Orazio Mirabella. Stochastic Analysis of Periodic Real-Time Systems in a Real-Time System. In *Proc. of the 23rd IEEE Real-Time Systems Symposium*, pages 289–300, Austin, Texas, December 2002.

[6] M. K. Gardner and J. W.S. Liu. Analyzing Stochastic Fixed-Priority Real-Time Systems. In *Proc. of the 5th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, Mar. 1999.

[7] Mark K. Gardner. *Probabilistic Analysis and Scheduling of Critical Soft Real-Time Systems*. PhD thesis, University of Illinois, Urbana-Champaign, 1999.

[8] J. P. Lehoczky. Real-Time Queueing Theory. In *Proc. of the 17th IEEE Real-Time Systems Symposium*, pages 186–195, Dec. 1996.

[9] J. P. Lehoczky. Real-Time Queueing Network Theory. In *Proc. of the 18th IEEE Real-Time Systems Symposium*, pages 58–67, Dec. 1997.

[10] John P. Lehoczky. Fixed Priority Scheduling of Periodic Task Sets with Arbitrary Deadlines. In *Proc. of the 11th IEEE Real-Time Systems Symposium*, pages 201–209, December 1990.

[11] L. Liu and J. Layland. Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment. *Journal of ACM*, 20(1):46–61, 1973.

[12] S. Manolache, P. Eles, and Z. Peng. Memory and Time-Efficient Schedulability Analysis of Task Sets with Stochastic Execution Times. In *Proc. of the 13th Euromicro Conference on Real-Time Systems*, pages 19–26, Jun. 2001.

[13] A. K. Mok and D. Chen. A Multiframe Model for Real-Time Tasks. *IEEE Transactions on Software Engineering*, 23(10):635–645, Oct. 1997.

[14] Lui Sha, Ragunathan Rajkumar, and John P. Lehoczky. Priority inheritance protocols: An approach to real-time synchronization. *IEEE Transactions on Computers*, 39(9):1175–1185, September 1990.

[15] T.-S. Tia, Z. Deng, M. Shankar, M. Storch, J. Sun, L.-C. Wu, and J.W.-S Liu. Probabilistic Performance Guarantee for Real-Time Tasks with Varying Computation Times. In *Proc. of the Real-Time Technology and Applications Symposium*, pages 164–173, Chicago, Illinois, May 1995.

[16] K. Tindell, A. Burns, and A. J. Wellings. An Extendible Approach for Analyzing Fixed Priority Hard Real-Time Tasks. *Real-Time Systems*, 6:133–151, 1994.